

Cahiers **GUT** *enberg*

♫ LE RAPPORT D'ACTIVITÉ DE L'INRIA

¶ José GRIMM

Cahiers GUTenberg, n° 32 (1999), p. 35-45.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1999__32_35_0>

© Association GUTenberg, 1999, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique

est constitutive d'une infraction pénale. Toute copie ou impression

de ce fichier doit contenir la présente mention de copyright.

Le rapport d'activité de l'Inria

José GRIMM

INRIA Sophia Antipolis
2004 Route des Lucioles
BP 93, F-06902 Sophia Antipolis CEDEX
jose.grimm@sophia.inria.fr

Résumé. Nous décrivons dans cet article comment le rapport d'activité de l'Inria, formé d'environ 80 documents \LaTeX indépendants est imprimé sous forme de neuf volumes papier (au total 2 294 pages A4), et traduit en html (3 131 pages *Web*) par `latex2html`. On utilise trois classes de document, trois fichiers de style de bibliographie, et deux script `perl`.

1. Introduction

Chaque année depuis sa création, l'Institut National de Recherche en Informatique et en Automatique (Inria) produit un rapport d'activité. Depuis 1987, ce rapport est composé en \LaTeX [4]. Celui-ci est traduit en html depuis 1994. La première année, on avait utilisé le logiciel `Gri f`, avec un passage via `sgml`, mais depuis, nous produisons directement du html via `latex2html`. Cet article décrit la stratégie utilisée en 1998, phase transitoire vers le *raweb*, qui pourrait utiliser `xml`, `mathml`, et autres outils en cours de développement.

L'Inria est un institut qui regroupe cinq unités de recherche en France. Il y a quatre thèmes de recherche, découpés en neuf programmes, et environ 80 projets. Le rapport Inria national est imprimé sous la forme de neuf volumes (un par programme), ce qui fait environ 2 200 pages. Chaque unité de recherche édite son propre tiré à part (contenant éventuellement des textes supplémentaires, certaines unités comme le Loria ou l'Irisa étant associées au CNRS et à l'Université).

Le problème est donc le suivant : étant donnés 80 documents \LaTeX , les traduire en html, et les imprimer de plusieurs façons différentes : en mode « brouillon », chaque texte séparément, et par volumes (collection d'un certain nombre de textes choisis suivant un critère prédéfini ou non).

2. Spécifications

Un certain nombre de contraintes syntaxiques et sémantiques sont imposées aux rédacteurs. Chaque texte est formé de dix sections ; la première section est formée de la composition de l'équipe, la dernière section est formée de la bibliographie. Les autres sections ont une structure commune : on y décrit les objectifs principaux de recherche, les résultats nouveaux, les contrats industriels, les logiciels, etc., sous forme de modules.

Chaque module correspond à une sous-section \LaTeX et une page *Web*. Il est formé de méta-données et d'un texte. Les méta-données sont : les participants à l'action de recherche, les mots clés, un glossaire, etc., tandis que le texte est formé d'un corps, précédé d'un résumé éventuel. Un module doit pouvoir être lu de façon indépendante des autres, mais peut faire référence à d'autres modules. Des liens hypertextes vers des pages *Web* sont fortement encouragés.

3. Structuration

Le rapport d'un projet a la forme suivante :

```
\documentclass{ra98}
\projet{MIAOU}{Miaou}{Mathématiques et Informatique de
l'Automatique et de l'Optimisation pour l'Utilisateur}
\typeprojet{Projet}
\localisation{Sophia Antipolis}
\theme{4a}
\newcommand\truc{truc}
\usepackage{xy}
\begin{document}
  \maketitle
  Ici se trouvent les modules.
  \loadbiblio
\end{document}
```

À l'Inria, chaque projet a un nom de code (ici Miaou), et un nom développé (par exemple « Mathématiques et Informatique... »). Le premier argument de la commande `\projet` est le nom de code en lettres majuscules non accentuées, le second est le vrai nom de code, et le dernier est le nom développé. Le nom du projet est suivi par un certain nombre d'informations (il s'agit d'un projet de recherche, cela pourrait être une action, un avant-projet,

etc., le projet est localisé à Sophia Antipolis et est dans le thème 4, programme 4a).

Avant le `\begin{document}`, on peut définir des commandes \LaTeX , et inclure des *packages*. L'utilisateur est obligé de rajouter au moins les deux commandes `\maketitle` et `\loadbiblio`: la raison en est que les commandes \LaTeX du type `\AtBeginDocument` ou `\AtEndDocument` ne fonctionnent pas comme nous l'aimerions. En particulier, `\AtBeginDocument{\foo}` exécute la commande `\foo` avant que le style *french* ne soit activé.

Un exemple de module est le suivant :

```
\begin{module}{logiciels}{hyperion}{Le logiciel {\sc hyperion}}
  \begin{participants}
    \pers{José}{Grimm}, \pers{Laurent}{Baratchart}[le chef]
  \end{participants}
  \begin{motscle} identification, approximation \end{motscle}
  \begin{abstract}
    Un petit résumé.
  \end{abstract}
  Description du logiciel.
\end{module}
```

L'environnement `module` prend trois arguments: le nom de la section dans laquelle est placé le module, le nom du module, et le titre du module. Le nom du projet, le nom de la section et le nom du module (les deux derniers étant formés uniquement caractères minuscules non accentués), identifient de façon unique le module, et donc la page *Web* associée. Comme `latex2html` refuse de générer des noms de fichiers ayant plus de 32 caractères, le nom de la section est tronqué sur 5 caractères, et le nom du module sur 21 caractères. Le module donné en exemple sera placé, après traduction, à l'URL suivante¹ :

http://www.inria.fr/Rapports/miaou/logic_dom-hyperion.html

Il est possible, via la commande `\moduleref{MIAOU}{logiciels}{hyperion}`, de référencer ce module. Il est prévu d'insérer un argument optionnel, qui est l'année, ce qui permettra de faire une référence à un module de 1998 en 1999 (sur le serveur *Web*, `Rapports` est un lien vers `RA98`, ce qui rend la transition automatique).

1. Par construction, ce nom ne comporte que des caractères ASCII standard. Nous avons triché un peu, car le nom du module dans le fichier source est `dom-hyperion`, l'URL citée plus haut existe effectivement.

Nous avons spécifié trois classes de documents :

- `ra98.cls` : cette classe permet d'imprimer un document, en mode brouillon, sans prétraitement ;
- `raweb.cls` : cette classe permet d'imprimer un document isolé dans sa forme définitive (le PostScript généré de cette manière est placé sur le serveur ftp de l'Inria). Ceci nécessite un prétraitement.
- `raglobal.cls` : cette classe permet de créer un document contenant un ensemble de textes (par exemple tous ceux de Sophia Antipolis). Il faut au préalable exécuter \LaTeX et \BibTeX sur chaque document.

4. Les classes de document

L'un des problèmes est la traduction de la commande `\moduleref`. Si les arguments sont x , y et z , la macro s'expande en `\ref{x@y@z}` si le document global contient le fichier x , et en « voir projet x , section y , module z^2 » dans le cas contraire. Dans la classe `ra98`, le test sur x est fait en comparant le nom du projet et une variable globale positionnée par la commande `\projet`. Dans les autres cas, la commande `\projet` génère un `\label` et on teste si ce label existe³.

Les deux classes `raweb` et `raglobal` nécessitent un prétraitement. Il s'agit d'un script `perl` qui effectue un certain nombre de tests, et qui découpe le source \LaTeX en $N + 4$ fichiers, où N est le nombre de modules. À chaque module est associé un fichier. Dans le cas du module cité plus haut, le fichier `miaou_logiciels_hyperion.tex` commence ainsi :

```
\startmodule{MIAOU}{6}{hyperion}
{\ignoresname{6/hyperion}Le logiciel {\sc hyperion}}
\ralabel{MIAOU@logiciels@hyperion}
```

Il est à noter que le nom de la section est parfois remplacé par un numéro (6, dans l'exemple). La commande `\ignoresname` ignore son argument, son intérêt sera décrit plus loin.

Le prétraitement génère un fichier principal de la forme :

```
\documentclass{raweb}
```

2. Certains modules ont un nom peu significatif, comme `module-17`, il faudrait pouvoir récupérer et imprimer le titre du module.

3. Une solution plus simple serait de tester l'existence du label `x@y@z`.

```

\input{miaou_dec11}
\begin{document}
\input{miaou_dec12}
\input{miaou_modules}
\end{document}

```

et trois fichiers auxiliaires. Dans le fichier *modules*, il y a un `\input` par module. Il y a aussi des commandes `\section{xx}`, où le titre de la section est obtenu à partir du nom symbolique via une table⁴.

Le fichier *decl1* contient le préambule du document source, à l'exception des informations obtenues via `\projet`, etc. Il contient donc uniquement des définitions de commandes \LaTeX , et des inclusions de *package*. Si l'on veut créer un document avec un ensemble de projets, il faut concaténer ces nouveaux préambules (un exemple typique de problème est que les projet définissent souvent une macro `\R` pour décrire l'ensemble des réels, et parfois de diverses façons). La phase de prétraitement vérifie qu'il n'y a aucune définition de commandes après le `\begin{document}`.

Pour résoudre ces conflits, on a le choix entre utiliser une version commune, (dans le fichier de classe) ou renommer cette macro, en la préfixant par le nom du projet. Une autre stratégie consiste à créer deux fichiers *debmac* et *finmac* qui contiennent respectivement le code des macros, et le code annulateur (du genre : `\let\R\relax` ou `\let\R\miaou@R` en cas de redéfinition). La classe *raweb* charge ces fichiers s'ils existent⁵.

Le fichier *decl2* contient tout le reste. Il contient également une page de titre spécifique pour le *Web* (voir figure 1). Cette page de titre est ignorée par \LaTeX , car incluse dans un environnement `htmlonly`.

La classe *raweb* est basée sur la classe *article* : la commande `\maketitle` de *decl2* produit une page de titre normalement. La classe *raglobal* est basée sur la classe *book*, la commande `\maketitle` y est redéfinie essentiellement comme `\chapter` (avec un en-tête de chapitre spécial). Cette classe gère essentiellement les conflits de noms potentiels entre les documents :

- comme indiqué plus haut, si deux projets définissent une même macro `\R`, un traitement manuel doit être exécuté ;
- les labels sont rendus uniques par ajout d'un préfixe (le nom du projet) aux arguments des commandes `\label` et `\ref`. Ce préfixe n'est pas rajouté dans les cas où il existe déjà (cas de `\ralabel` et `\moduleref`) ;

4. Par exemple « Logiciels » est associé à « logiciels ».

5. On pense automatiser cela l'année prochaine.

Projet : MIAOU



Suivant : Composition de l'équipe

**Mathématiques et Informatique de
l'Automatique et de l'Optimisation pour
l'Utilisateur**

MIAOU

Rapport d'activité de l'année 1998

Sophia Antipolis

Thème : 4a

Page de présentation du projet - Rapport d'activité au format PostScript ou PDF

- Composition de l'équipe
 - Responsable scientifique
 - Responsable permanent
 - Assistante de projet
 - Personnel INRIA
 - Chercheurs doctorants
 - Chercheur post-doctorant
 - Conseillers scientifiques
 - Chercheurs invités
 - Stagiaires
- Présentation et objectifs généraux
- Fondements scientifiques

FIG. 1 – Exemple de page de titre pour le projet Miaou. Nous n'avons indiqué que le début de la table des matières.

- le même préfixe est rajouté aux commandes de citations (les commandes internes associés à `\cite` et `\bibitem`)⁶ ;
- on suppose que tous les fichiers du projet x sont dans un répertoire x . Ce préfixe est rajouté à tous les arguments de la commande `\input`, et, pour pouvoir inclure des images, dans la commande `\graphicspath`⁷.

5. Génération du code html

Le code html est généré via `latex2html`, qui est un script perl paramétrable de plusieurs façons ; nous les utilisons toutes. Comme indiqué plus haut, on utilise l'environnement `htmlonly` pour la page de titre.

Nous avons redéfini la fonction `custom_title_hook`, c'est elle qui calcule le nom de la page *Web*. Elle reçoit en argument le titre, et l'identification de la page ; il s'agit d'une suite de nombres (numéros de partie, chapitre, section, sous-section, etc.). Seuls les numéros de section et de sous-section nous intéressent. Si le numéro de sous-section est nul, il s'agit d'une section⁸. Le nom de cette section est obtenu via une table (par exemple, la section 6 est la section « logiciels »). Notons que si la section 4 est vide, la section logiciels est numérotée 5. Dans la phase de prétraitement, une seconde table d'indirection est calculée⁹, et est placée dans le fichier `.latex2htmlinit`. Si le numéro de sous-section est non nul, le nom du module est obtenu via une variable globale, calculée lors du traitement de la commande `\ignorename`.

Dans le fichier `raweb.perl`, on définit un certain nombre de procédure perl, une par commande ou environnement L^AT_EX, comme par exemple la commande `\pers` et l'environnement `participants`. Un traitement spécifique est fait pour les commandes définies par le style *french*, par exemple `\ier`. Notons que le style *french* remplace les double inférieur ou double supérieur par des guillemets. Ceci est également fait, via la procédure `pre_pre_process`, qui est appelée sur le texte brut. Une des difficultés non résolues est que, alors que le style *french* insère des espaces insécables automatiquement là où la typographie française l'exige, ni `latex2html` ni le script ne le font, et l'on retrouve parfois des deux-points en début de ligne (notre seul moyen d'action est sur le texte brut, nous ne voulons pas ajouter d'espaces dans les formules de math, par exemple).

6. On ne fait pas passer BIB_TE_X sur le résultat, cela simplifie la redéfinition des commandes.

7. Ceci interdit l'utilisation explicite de cette commande, et implique l'utilisation de la commande `\includegraphics` pour inclure des images.

8. Si le numéro de section est nul, c'est la page principale, son nom est le nom du projet.

9. Elle associe les numéros de section effectifs et les numéros attendus.

Le traitement de la commande `\moduleref` est simple : dans le cas où il s'agit du même projet, on insère un `\ref`, et dans le cas contraire, après troncatures adéquates, un `\htmladdnormallink`.

Un certain nombre d'autres traitements sont effectués pour pallier limitations de `latex2html` ; par exemple, on remplace œ par oe¹⁰.

6. La bibliographie

Chaque projet fournit trois fichiers de bibliographie : un fichier normal (publications de l'année), un fichier contenant les publications majeures du projet, et un fichier de références en note de bas de pages. Ces dernières sont insérées via le *package* `footbib`. Les publications de l'année sont classées par type (un fichier de style `raweb.bst` insère un code devant la clé, et on laisse `BIBTEX` trier ceci par ordre alphabétique). La phase de prétraitement concatène les trois fichiers `bb1` ainsi générés, pour en faire un seul. Les citations en note de bas de page sont ainsi remplacées par des citations normales dans le document *Web* (et le traducteur vers HTML ne voit qu'un seul fichier de bibliographie).

7. Extensions futures

L'idée est de créer dynamiquement un document papier à partir d'un certain nombre de modules choisis selon un certain critère (par exemple via une recherche sur les méta-données des modules). La sélection peut se faire via le *Web* : on proposera un *caddy*, dans lequel l'internaute mettra les pages *Web* sélectionnées. Le résultat sera la concaténation des fichiers `pdf` associés aux modules ; ce mécanisme devrait être disponible à l'été 1999. En interne à l'Inria, la méthode sera un peu différente : le résultat sera un document `LATEX`, contenant un ensemble de `\inputmodules`, et des commandes de présentation (le compilateur pourra rajouter une introduction, une conclusion, des `\section`, etc.). L'un des problèmes est alors la traduction des commandes `\ref`. Il faudra modifier la commande `\label` de telle sorte qu'elle imprime dans un fichier auxiliaire le nom du module, le nom du projet, le type de référence, le label et la page. Ainsi, `\ref{truc}` pourra se traduire en : « figure 17, page 103, volume 2A du rapport d'activité de l'Inria 1998 », à partir de la concaténation des fichiers auxiliaires associés aux divers volumes¹¹.

10. Une solution satisfaisante ne sera trouvée que lorsque ce caractère existera dans les polices standard.

11. Quid de : comparer les figures `\ref{truc}` et `\ref{trac}`?

Projet : MIAOU



Précédent : Logiciels **Remonter :** Logiciels **Suivant :** Un package pour la manipulation

Le logiciel hyperion

Participants : José Grimm [correspondant], Fabien Seyfert, Franck Wielonsky.

Le logiciel hyperion est un logiciel d'identification de fonctions de transfert multivariées, linéaires, stables et rationnelles, à partir de mesures fréquentielles partielles. Pour plus de détails, voir le rapport d'activité de l'année 1997, et la référence [[23]]. Ce logiciel est actuellement utilisé par le CNES et l'IRCOM (voir section 7.1). Notons que le CNES a acheté le droit d'utiliser le logiciel hyperion, et que l'IRCOM l'utilise aussi pour la synthèse de filtres diélectriques[BBA⁺].

Pour identifier une fonction F , on procède dans une première phase à sa « complétion analytique ». Dans cette phase, on essaie également d'identifier les retards éventuels (cf. section 6.6) et la valeur à l'infini (cf. section 6.5). Ensuite, on utilise la décomposition de Douglas-Shapiro-Shields $F = Q^{-1}C$, et l'on minimise une fonction $\psi(Q)$, où Q est une matrice intérieure de degré de McMillan donné. La fonction $\psi(Q)$, ainsi que ses dérivées première et seconde, sont calculées par application des techniques de différentiation automatique développées dans le projet SAFIR. Dans le cas scalaire, nous avons développé une version où la distance entre F et $Q^{-1}C$ n'est pas la distance L^2 usuelle, mais une distance pondérée (cf. section 6.4).

Nous avons également ajouté une extension à hyperion, qui utilise des propriétés physiques et géométriques particulières des filtres hyperfréquence, pour obtenir une réalisation de la fonction de transfert dans laquelle se lisent les paramètres physiques du filtre. Cette extension est codéveloppée avec le CNES.



Précédent : Logiciels **Remonter :** Logiciels **Suivant :** Un package pour la manipulation

FIG. 2 – Exemple de page Web générée. Noter les doubles crochets autour de la référence [[23]], l'espace manquant pour la référence en note de bas de page [BBA⁺], et la coupure dans la formule $F = Q^{-1}C$.

8. Problèmes non résolus

Certains problèmes se présentent lors de la génération du document \LaTeX global. Par exemple, certains auteurs utilisent la commande \TeX `\pmatrix`, alors que d'autres utilisent le *package* `amsmath`. Le mélange des deux provoque une erreur bizarre. Par ailleurs, la classe *raweb* utilise un corps 11 points, alors que la classes *raglobal* utilise un corps 10 points (pour gagner de la place). Les commandes de coupure de page introduites par les auteurs sont alors mal placées. Il en est de même des coupures de ligne et de mots. Ces dernières commandes nuisent relativement peu à la qualité du texte. Cependant, des commandes de césure apparaissent fréquemment dans les noms propres (particulièrement les noms étrangers), ce qui pose problème si l'on veut faire un index¹².

La plupart des problèmes se rencontrent cependant lors de la génération des pages *Web*. Ces problèmes sont souvent liés aux limitations des traducteurs. Par exemple, l'argument de `\vspace{0.30cm}` est traduit par `latex2html` en 30 cm¹³, ce qui donne un espace vertical déraisonnable dans le document. Une commande du type $\$x \text{\hbox{ si }} y\$$ provoque des résultats imprévisibles. En général, il suffit de remplacer la primitive \TeX par la commande \LaTeX adéquate (`\mbox` en l'occurrence). Le non respect de la syntaxe \LaTeX (par exemple `\mathbb X` au lieu de `\mathbb{X}`) peut provoquer des erreurs plus ou moins faciles à détecter¹⁴. Il est indispensable de relire soigneusement le document HTML généré.

Le traitement des images n'est pas toujours très satisfaisant : dans certains cas, \LaTeX positionne deux images côte à côte, alors que `latex2html` les place verticalement, ce qui rend une légende du type « comparer les images de gauche et de droite » un peu incompréhensible. Demander de tourner une image PostScript d'un quart de tour est parfois catastrophique : le calcul de la *bounding box* par `dvips` peut être tellement faux que l'image disparaît complètement.

Finalement, le traitement des URL pose quelques problèmes. Malgré toutes nos recommandations, certaines personnes n'utilisent pas les commandes du type `\htmladdnormallink`, d'autres pensent que la commande `\url` crée un lien hypertexte, de telle sorte qu'une vérification manuelle est nécessaire. Par ailleurs un grand nombre de liens sont invalides : il peut s'agir d'une faute de

12. En fait, on utilise de toutes façons un script `perl` pour faire l'index, car `makeindex` ne sait pas trier des mots avec des caractères accentués.

13. \TeX interprète évidemment ceci comme 3 mm.

14. Le message « Unknown commands: < » n'est pas dû à une commande `\<` non reconnue, mais à un backslash suivi d'un retour chariot au lieu d'une espace.

frappe, ou, malheureusement, d'un lien qui a disparu entre le moment de la rédaction et le traitement du document.

9. Conclusion

L'impression du rapport d'activité de l'Inria en volumes, et sa mise sur le *Web* sous la forme HTML est un travail de longue haleine. Nous avons mis en place un certain nombre d'outils destinés à automatiser le processus, et à contourner les problèmes de `latex2html`.

L'auteur tient à remercier tous les participants au rapport d'activité, et plus particulièrement Marie-Pierre Durollet qui a fait la traduction des textes en HTML, et Laurent Pierron qui a fait les volumes papier.

Bibliographie

- [1] Nikos DRAKOS *et al.* — *The latex2html Translator* (document électronique), 1998.
- [2] B. GAULLE. — *Utilisation du style french* (documentation du logiciel, v. 4,05b), 1999.
- [3] M. GOOSSENS, F. MITTELBACH & A. SAMARIN. — *The L^AT_EX companion*, Addison-Wesley, 1993.
- [4] Philippe LOUARN. — « Une expérience d'utilisation de L^AT_EX : le rapport d'activité de l'Inria », *Cahiers GUTenberg*, 0, avril 1988, p. 17-24.
- [5] V. QUINT & I. VATTON. — « Grif : un éditeur interactif de documents structurés », *Technical Report RR TIGRE 27*, IMAG, 1985.
- [6] Larry WALL, Tom CHRISTIANSEN, & Randal L. SCHWARTZ. — *Programming Perl*, O'Reilly & Associates, Inc, 1996.